CS15210: Modes and Media

Helen Miles (hem23@aber.ac.uk)

01/02/16

(based on slides by Mike Clarke)

Previously, in CS15210...

- How to do basic calculations for waves
    - use the indices tricks
- Different SI units:
    - giga (G), mega (M), kilo (k), milli (m), micro (μ), nano (n), ...
- Binary signals are a specific digital signal with only two values
    - don't have to be $\{1, 0\}$

Contents

Bits and Bytes: a Recap!

$$\boxed{1\ 0\ 1\ 1\ 0\ 0\ 1\ 0}$$

- A bit: single binary digit
- A byte: a group of bits (usually 8)

- Bits and bytes can be interpreted in many different ways
- In CS15210 we will usually interpret them as **ASCII** characters

ASCII

- ASCII: American Standard Code for Information Interchange
  - Defined by the American National Standards Institute (ANSI)
  - `http://www.asciitable.com/`
- A standard coding system that assigns a **7-bit** code to each letter, digit, and punctuation character
  - Separate codes for upper and lower case, e.g.
    A has ASCII code 65 (1000001),
    a has ASCII code 97 (1100001)
- This means we have a spare bit
  - We add a parity bit to check against errors

Parity

- A method to detect 1 bit errors in data
- Different types:
  - vertical/longitudinal parity
  - odd/even parity
- Procedure:
  - The transmitter and receiver must have agreed on a rule beforehand
  - The parity bit is added to the data at the transmission end
  - The receiver uses the rule to work out whether there's an error

Vertical parity

Whether there is an odd/even number of 1s in a byte

- The parity bit is added to correctly complete the byte and obey the rule being used
  - **Odd parity**: there must be an odd number of 1s in the byte
  - **Even parity**: there must be an even number of 1s in the byte

E.g. B has ASCII code 1000010

Odd parity: ?1000010

Even parity: ?1000010

Vertical parity

Whether there is an odd/even number of 1s in a byte

- The parity bit is added to correctly complete the byte and obey the rule being used
  - **Odd parity**: there must be an odd number of 1s in the byte
  - **Even parity**: there must be an even number of 1s in the byte

E.g. *B* has ASCII code ?1000010

Odd parity: 11000010

Even parity: 01000010

Vertical parity

Character	ASCII Code	Even Parity	Odd Parity
B	?1000010	01000010	11000010
b	?1100010	11100010	01100010
0	?0110000	?0110000	?0110000
1	?0110001	?0110001	?0110001
?	?0111111	?0111111	?0111111

Vertical parity

Character	ASCII Code	Even Parity	Odd Parity
B	?1000010	01000010	11000010
b	?1100010	11100010	01100010
0	?0110000	00110000	10110000
1	?0110001	10110001	00110001
?	?0111111	00111111	10111111

Vertical parity

On it's own, vertical parity can't identify which individual bit of a byte is incorrect

e.g. using even parity, 10111111 is wrong...
but we don't know which bit is wrong!

11111111, 10011111, 10111101, 00111111, ... ?

Enter longitudinal parity...

Longitudinal Parity

Parity can also be applied to the same bit in each byte of a block

1	0	**1**
0	0	**0**
1	0	**1**
1	1	**0**
0	1	**1**
0	0	**0**
1	0	**1**
0	0	**0**

Here we see 2 bytes of data being transmitted,
and an extra byte to check across the block

Longitudinal Parity

This is an example of even longitudinal parity
(or longitudinal redundancy check)

1	0	**1**	$\leftarrow 2$
0	0	**0**	$\leftarrow 0$
1	0	**1**	$\leftarrow 2$
1	1	**0**	$\leftarrow 0$
0	1	**1**	$\leftarrow 2$
0	0	**0**	$\leftarrow 0$
1	0	**1**	$\leftarrow 2$
0	0	**0**	$\leftarrow 0$
$\uparrow$	$\uparrow$	$\uparrow$	
3	2	4	

Longitudinal Parity

The combination of longitudinal and vertical parity means that single bit errors can be detected and corrected

$$
\begin{array}{ccc|c}
1 & 0 & \mathbf{1} & \leftarrow 2 \\
0 & 0 & \mathbf{0} & \leftarrow 0 \\
0 & 0 & \mathbf{1} & \leftarrow 1 \\
1 & 1 & \mathbf{0} & \leftarrow 0 \\
0 & 1 & \mathbf{1} & \leftarrow 2 \\
0 & 0 & \mathbf{0} & \leftarrow 0 \\
1 & 0 & \mathbf{1} & \leftarrow 2 \\
0 & 0 & \mathbf{0} & \leftarrow 0 \\
\hline
\uparrow & \uparrow & \uparrow & \\
3 & 2 & 4 &
\end{array}
$$

Example: 16 bytes, where is the error?

- Odd longitudinal parity – rows contain an odd number of 1s
- Even vertical parity – columns contain an even number of 1s

$$
\begin{array}{ccccccccccccccccc|c}
1&0&1&1&0&0&1&1&0&1&1&0&1&1&1&1&\mathbf{0}&\leftarrow\\
0&0&1&1&1&1&0&0&1&1&1&1&0&1&0&0&\mathbf{0}&\leftarrow\\
1&0&1&0&1&0&1&1&0&0&0&0&0&0&0&1&\mathbf{1}&\leftarrow\\
1&1&1&1&1&1&1&1&1&0&0&0&0&1&1&0&\mathbf{0}&\leftarrow\\
0&1&1&1&1&1&0&1&1&0&0&0&0&1&1&1&\mathbf{0}&\leftarrow\\
0&0&0&1&1&0&1&0&1&1&1&0&0&1&0&0&\mathbf{0}&\leftarrow\\
1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&\mathbf{0}&\leftarrow\\
\color{red}0&\color{red}0&\color{red}1&\color{red}1&\color{red}1&\color{red}0&\color{red}0&\color{red}1&\color{red}0&\color{red}1&\color{red}1&\color{red}1&\color{red}1&\color{red}1&\color{red}1&\color{red}0&\color{red}\mathbf{1}&\leftarrow\\
\hline
\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&
\end{array}
$$

Example: 16 bytes, where is the error?

- Odd longitudinal parity – rows contain an odd number of 1s
- Even vertical parity – columns contain an even number of 1s

1	0	1	1	0	0	1	1	0	1	1	0	1	1	1	1	**0**	← 11
0	0	1	1	1	1	0	0	1	1	1	1	0	1	0	0	**0**	←
1	0	1	0	1	0	1	1	0	0	0	0	0	0	0	1	**1**	←
1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	0	**0**	←
0	1	1	1	1	1	0	1	1	0	0	0	0	1	1	1	**0**	←
0	0	0	1	1	0	1	0	1	1	1	0	0	1	0	0	**0**	←
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	**0**	←
0	0	1	1	1	0	0	1	0	1	1	1	1	1	1	0	**1**	←
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	

Example: 16 bytes, where is the error?

- Odd longitudinal parity – rows contain an odd number of 1s
- Even vertical parity – columns contain an even number of 1s

$$
\begin{array}{ccccccccccccccccc|l}
1&0&1&1&0&0&1&1&0&1&1&0&1&1&1&1&\mathbf{0} & \leftarrow 11\\
0&0&1&1&1&1&0&0&1&1&1&1&0&1&0&0&\mathbf{0} & \leftarrow 9\\
1&0&1&0&1&0&1&1&0&0&0&0&0&0&0&1&\mathbf{1} & \leftarrow\\
1&1&1&1&1&1&1&1&1&0&0&0&0&1&1&0&\mathbf{0} & \leftarrow\\
0&1&1&1&1&1&0&1&1&0&0&0&0&1&1&1&\mathbf{0} & \leftarrow\\
0&0&0&1&1&0&1&0&1&1&1&0&0&1&0&0&\mathbf{0} & \leftarrow\\
1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&\mathbf{0} & \leftarrow\\
0&0&1&1&1&0&0&1&0&1&1&1&1&1&1&0&\mathbf{1} & \leftarrow\\
\hline
\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow &
\end{array}
$$

Example: 16 bytes, where is the error?

- Odd longitudinal parity – rows contain an odd number of 1s
- Even vertical parity – columns contain an even number of 1s

$$
\begin{array}{ccccccccccccccccc|l}
1&0&1&1&0&0&1&1&0&1&1&0&1&1&1&1&\mathbf{0} & \leftarrow 11 \\
0&0&1&1&1&1&0&0&1&1&1&1&0&1&0&0&\mathbf{0} & \leftarrow 9 \\
1&0&1&0&1&0&1&1&0&0&0&0&0&0&0&1&\mathbf{1} & \leftarrow 7 \\
1&1&1&1&1&1&1&1&1&0&0&0&0&1&1&0&\mathbf{0} & \leftarrow \\
0&1&1&1&1&1&0&1&1&0&0&0&0&1&1&1&\mathbf{0} & \leftarrow \\
0&0&0&1&1&0&1&0&1&1&1&0&0&1&0&0&\mathbf{0} & \leftarrow \\
1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&\mathbf{0} & \leftarrow \\
\color{red}0&\color{red}0&\color{red}1&\color{red}1&\color{red}1&\color{red}0&\color{red}0&\color{red}1&\color{red}0&\color{red}1&\color{red}1&\color{red}1&\color{red}1&\color{red}1&\color{red}1&\color{red}0&\color{red}\mathbf{1} & \leftarrow \\
\hline
\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow &
\end{array}
$$

Example: 16 bytes, where is the error?

- Odd longitudinal parity – rows contain an odd number of 1s
- Even vertical parity – columns contain an even number of 1s

1	0	1	1	0	0	1	1	0	1	1	0	1	1	1	1	**0**	$\leftarrow$ 11
0	0	1	1	1	1	0	0	1	1	1	1	0	1	0	0	**0**	$\leftarrow$ 9
1	0	1	0	1	0	1	1	0	0	0	0	0	0	0	1	**1**	$\leftarrow$ 7
1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	0	**0**	$\leftarrow$ 11
0	1	1	1	1	1	0	1	1	0	0	0	0	1	1	1	**0**	$\leftarrow$
0	0	0	1	1	0	1	0	1	1	1	0	0	1	0	0	**0**	$\leftarrow$
1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	**0**	$\leftarrow$
0	0	1	1	1	0	0	1	0	1	1	1	1	1	1	0	**1**	$\leftarrow$
$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	

Example: 16 bytes, where is the error?

- Odd longitudinal parity – rows contain an odd number of 1s
- Even vertical parity – columns contain an even number of 1s

$$
\begin{array}{ccccccccccccccccc|l}
1&0&1&1&0&0&1&1&0&1&1&0&1&1&1&1&\mathbf{0}&\leftarrow 11\\
0&0&1&1&1&1&0&0&1&1&1&1&0&1&0&0&\mathbf{0}&\leftarrow 9\\
1&0&1&0&1&0&1&1&0&0&0&0&0&0&0&1&\mathbf{1}&\leftarrow 7\\
1&1&1&1&1&1&1&1&1&0&0&0&0&1&1&0&\mathbf{0}&\leftarrow 11\\
\color{blue}0&\color{blue}1&\color{blue}1&\color{blue}1&\color{blue}1&\color{blue}1&\color{blue}0&\color{blue}1&\color{blue}1&\color{blue}0&\color{blue}0&\color{blue}0&\color{blue}0&\color{blue}1&\color{blue}1&\color{blue}1&\color{blue}\mathbf{0}&\color{blue}\leftarrow 10\\
0&0&0&1&1&0&1&0&1&1&1&0&0&1&0&0&\mathbf{0}&\leftarrow\\
1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&\mathbf{0}&\leftarrow\\
\color{red}0&\color{red}0&\color{red}1&\color{red}1&\color{red}1&\color{red}0&\color{red}0&\color{red}1&\color{red}0&\color{red}1&\color{red}1&\color{red}1&\color{red}1&\color{red}1&\color{red}1&\color{red}0&\color{red}\mathbf{1}&\color{red}\leftarrow\\
\hline
\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&
\end{array}
$$

Example: 16 bytes, where is the error?

- Odd longitudinal parity – rows contain an odd number of 1s
- Even vertical parity – columns contain an even number of 1s

1	0	1	1	0	0	1	1	0	1	1	0	1	1	1	1	**0**	← 11
0	0	1	1	1	1	0	0	1	1	1	1	0	1	0	0	**0**	← 9
1	0	1	0	1	0	1	1	0	0	0	0	0	0	0	1	**1**	← 7
1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	0	**0**	← 11
0	1	1	1	1	1	0	1	1	0	0	0	0	1	1	1	**0**	← 10
0	0	0	1	1	0	1	0	1	1	1	0	0	1	0	0	**0**	← 7
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	**0**	← 3
0	0	1	1	1	0	0	1	0	1	1	1	1	1	1	0	**1**	← 11
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	

Example: 16 bytes, where is the error?

- Odd longitudinal parity – rows contain an odd number of 1s
- Even vertical parity – columns contain an even number of 1s

1	0	1	1	0	0	1	1	0	1	1	0	1	1	1	1	**0**	← 11
0	0	1	1	1	1	0	0	1	1	1	1	0	1	0	0	**0**	← 9
1	0	1	0	1	0	1	1	0	0	0	0	0	0	0	1	**1**	← 7
1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	0	**0**	← 11
0	1	1	1	1	1	0	1	1	0	0	0	0	1	1	1	**0**	← 10
0	0	0	1	1	0	1	0	1	1	1	0	0	1	0	0	**0**	← 7
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	**0**	← 3
0	0	1	1	1	0	0	1	0	1	1	1	1	1	1	0	**1**	← 11
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	
4																	

Example: 16 bytes, where is the error?

- Odd longitudinal parity – rows contain an odd number of 1s
- Even vertical parity – columns contain an even number of 1s

1	0	1	1	0	0	1	1	0	1	1	0	1	1	1	1	**0**	← 11
0	0	1	1	1	1	0	0	1	1	1	1	0	1	0	0	**0**	← 9
1	0	1	0	1	0	1	1	0	0	0	0	0	0	0	1	**1**	← 7
1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	0	**0**	← 11
0	1	1	1	1	1	0	1	1	0	0	0	0	1	1	1	**0**	← 10
0	0	0	1	1	0	1	0	1	1	1	0	0	1	0	0	**0**	← 7
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	**0**	← 3
0	0	1	1	1	0	0	1	0	1	1	1	1	1	1	0	**1**	← 11
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	
4	2																

Example: 16 bytes, where is the error?

- Odd longitudinal parity – rows contain an odd number of 1s
- Even vertical parity – columns contain an even number of 1s

1	0	1	1	0	0	1	1	0	1	1	0	1	1	1	1	**0**	← 11
0	0	1	1	1	1	0	0	1	1	1	1	0	1	0	0	**0**	← 9
1	0	1	0	1	0	1	1	0	0	0	0	0	0	0	1	**1**	← 7
1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	0	**0**	← 11
0	1	1	1	1	1	0	1	1	0	0	0	0	1	1	1	**0**	← 10
0	0	0	1	1	0	1	0	1	1	1	0	0	1	0	0	**0**	← 7
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	**0**	← 3
0	0	1	1	1	0	0	1	0	1	1	1	1	1	1	0	**1**	← 11
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	
4	2	6															

Example: 16 bytes, where is the error?

- Odd longitudinal parity – rows contain an odd number of 1s
- Even vertical parity – columns contain an even number of 1s

1	0	1	1	0	0	1	1	0	1	1	0	1	1	1	1	**0**	← 11
0	0	1	1	1	1	0	0	1	1	1	1	0	1	0	0	**0**	← 9
1	0	1	0	1	0	1	1	0	0	0	0	0	0	0	1	**1**	← 7
1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	0	**0**	← 11
0	1	1	1	1	1	0	1	1	0	0	0	0	1	1	1	**0**	← 10
0	0	0	1	1	0	1	0	1	1	1	0	0	1	0	0	**0**	← 7
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	**0**	← 3
0	0	1	1	1	0	0	1	0	1	1	1	1	1	1	0	**1**	← 11
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	
4	2	6	6														

Example: 16 bytes, where is the error?

- Odd longitudinal parity – rows contain an odd number of 1s
- Even vertical parity – columns contain an even number of 1s

1	0	1	1	0	0	1	1	0	1	1	0	1	1	1	1	**0**	← 11
0	0	1	1	1	1	0	0	1	1	1	1	0	1	0	0	**0**	← 9
1	0	1	0	1	0	1	1	0	0	0	0	0	0	0	1	**1**	← 7
1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	0	**0**	← 11
0	1	1	1	1	1	0	1	1	0	0	0	0	1	1	1	**0**	← 10
0	0	0	1	1	0	1	0	1	1	1	0	0	1	0	0	**0**	← 7
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	**0**	← 3
0	0	1	1	1	0	0	1	0	1	1	1	1	1	1	0	**1**	← 11
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	
4	2	6	6	6													

Example: 16 bytes, where is the error?

- Odd longitudinal parity – rows contain an odd number of 1s
- Even vertical parity – columns contain an even number of 1s

1	0	1	1	0	0	1	1	0	1	1	0	1	1	1	1	**0**	← 11
0	0	1	1	1	1	0	0	1	1	1	1	0	1	0	0	**0**	← 9
1	0	1	0	1	0	1	1	0	0	0	0	0	0	0	1	**1**	← 7
1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	0	**0**	← 11
0	1	1	1	1	1	0	1	1	0	0	0	0	1	1	1	**0**	← 10
0	0	0	1	1	0	1	0	1	1	1	0	0	1	0	0	**0**	← 7
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	**0**	← 3
0	0	1	1	1	0	0	1	0	1	1	1	1	1	1	0	**1**	← 11
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	
4	2	6	6	6	3												

Example: 16 bytes, where is the error?

- Odd longitudinal parity – rows contain an odd number of 1s
- Even vertical parity – columns contain an even number of 1s

$$
\begin{array}{ccccccccccccccccc|l}
1&0&1&1&0&0&1&1&0&1&1&0&1&1&1&1&\mathbf{0} & \leftarrow 11\\
0&0&1&1&1&1&0&0&1&1&1&1&0&1&0&0&\mathbf{0} & \leftarrow 9\\
1&0&1&0&1&0&1&1&0&0&0&0&0&0&0&1&\mathbf{1} & \leftarrow 7\\
1&1&1&1&1&1&1&1&1&0&0&0&0&1&1&0&\mathbf{0} & \leftarrow 11\\
0&1&1&1&1&1&0&1&1&0&0&0&0&1&1&1&\mathbf{0} & \leftarrow 10\\
0&0&0&1&1&0&1&0&1&1&1&0&0&1&0&0&\mathbf{0} & \leftarrow 7\\
1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&\mathbf{0} & \leftarrow 3\\
0&0&1&1&1&0&0&1&0&1&1&1&1&1&1&0&\mathbf{1} & \leftarrow 11\\
\hline
\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow&\uparrow & \\
4&2&6&6&6&3&4&6&4&4&4&2&2&6&4&4&\mathbf{2} &
\end{array}
$$

Parity Checking

What if multiple bits are wrong?

0	0	1	1	0	0	1	1	0	1	1	0	1	1	1	1	**0**	← 10
0	0	1	1	1	1	0	0	1	1	1	1	0	1	0	0	**0**	← 9
1	0	1	0	1	0	1	1	0	0	0	0	1	0	0	1	**1**	← 8
1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	0	**0**	← 11
0	1	1	1	1	1	0	1	1	0	0	0	0	1	1	1	**0**	← 10
0	0	0	1	1	0	1	0	1	1	1	0	0	1	0	0	**0**	← 7
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	**0**	← 3
0	0	1	1	1	0	0	0	0	1	1	1	1	1	1	0	**1**	← 10
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	
5	2	6	6	6	3	4	5	4	4	4	2	3	6	4	4	**2**	

Parity Checking

May not always work...

$$
\begin{array}{ccccccccccccccccc|l}
1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & \mathbf{0} & \leftarrow 11 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & \mathbf{0} & \leftarrow 8 \\
1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \mathbf{1} & \leftarrow 8 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & \mathbf{0} & \leftarrow 10 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & \mathbf{0} & \leftarrow 10 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & \mathbf{0} & \leftarrow 7 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \mathbf{0} & \leftarrow 3 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & \mathbf{1} & \leftarrow 11 \\
\hline
\uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \\
4 & 2 & 6 & 6 & 6 & 2 & 4 & 6 & 4 & 4 & 4 & 2 & 2 & 6 & 4 & 4 & \mathbf{2} &
\end{array}
$$

Parity Checking

May not always work...

$$
\begin{array}{ccccccccccccccccc|l}
1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & \mathbf{0} & \leftarrow 11 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & \mathbf{0} & \leftarrow 8 \\
1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \mathbf{1} & \leftarrow 8 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & \mathbf{0} & \leftarrow 10 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & \mathbf{0} & \leftarrow 10 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & \mathbf{0} & \leftarrow 7 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \mathbf{0} & \leftarrow 3 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & \mathbf{1} & \leftarrow 11 \\
\hline
\uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \\
4 & 2 & 6 & 6 & 6 & 2 & 4 & 6 & 4 & 4 & 4 & 2 & 2 & 6 & 4 & 4 & \mathbf{2} &
\end{array}
$$

Parity Checking

May not always work...

$$
\begin{array}{ccccccccccccccccc|c}
1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & \mathbf{0} & \leftarrow 11 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \mathbf{0} & \leftarrow 7 \\
1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & \mathbf{1} & \leftarrow 9 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & \mathbf{0} & \leftarrow 11 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & \mathbf{0} & \leftarrow 11 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & \mathbf{0} & \leftarrow 7 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \mathbf{0} & \leftarrow 3 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & \mathbf{1} & \leftarrow 11 \\
\hline
\uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \\
4 & 2 & 6 & 6 & 6 & 2 & 4 & 6 & 4 & 6 & 4 & 2 & 2 & 6 & 4 & 4 & \mathbf{2} &
\end{array}
$$

Parity Checking

May not always work...

1	0	1	1	0	0	1	1	0	1	1	0	1	1	1	1	**0**	← 11
0	0	1	1	1	0	0	0	1	0	1	1	0	1	0	0	**0**	← 7
1	0	1	0	1	1	1	1	0	1	0	0	0	0	0	1	**1**	← 9
1	1	1	1	1	0	1	1	1	1	0	0	0	1	1	0	**0**	← 11
0	1	1	1	1	1	0	1	1	1	0	0	0	1	1	1	**0**	← 11
0	0	0	1	1	0	1	0	1	1	1	0	0	1	0	0	**0**	← 7
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	**0**	← 3
0	0	1	1	1	0	0	1	0	1	1	1	1	1	1	0	**1**	← 11
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	
4	2	6	6	6	2	4	6	4	6	4	2	2	6	4	4	**2**	

Parity

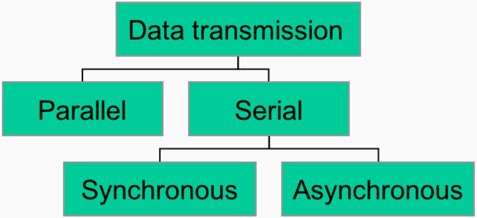
- A method to detect 1 bit errors in data
- May work for multiple bit errors
- Binary errors are easily corrected
  - if it's wrong it must be the other value
- Different types:
  - vertical/longitudinal parity
  - odd/even parity
- The transmitter and receiver must have agreed on a rule
- The receiver uses the rule to work out whether there's an error

Transmission Modes

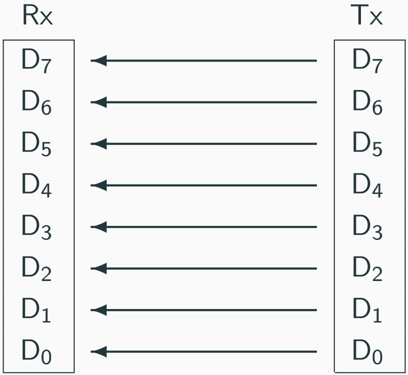
- **simplex** mode means that data can flow in one direction only, e.g. a public address system, tannoys
- **half-duplex** means that data can flow in both directions but not at the same time
- **full-duplex** means that data can flow in both directions at the same time

(Note that there is some confusion over these terms. The above is American usage, which is what we will use.)

Types of Data Transmission



Parallel Transmission

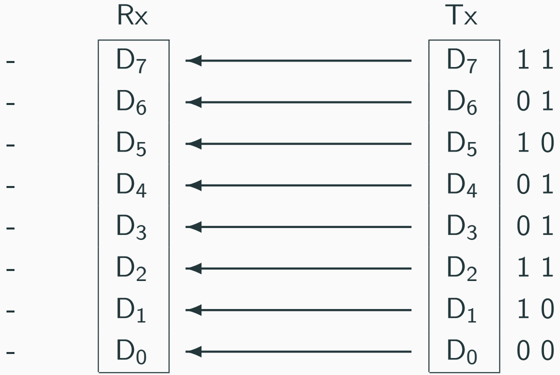


A number of bits can be transmitted simultaneously by allowing a wire for each bit

Parallel Transmission

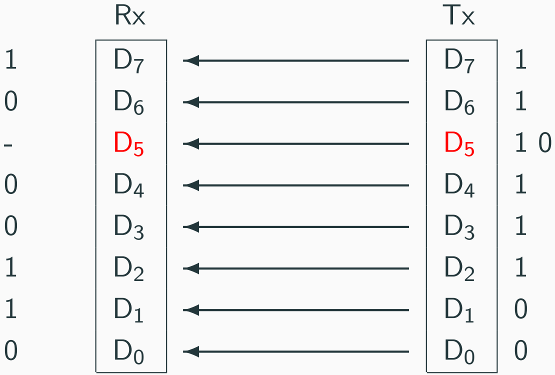
- A number of bits (usually a byte, i.e. 8 bits) are transmitted simultaneously
- This is fast and convenient
- But expensive, because we need eight wires
- Limited to very short distances ($\approx 10\,\mathrm{m}$) because of the problem of skewing

Parallel Transmission



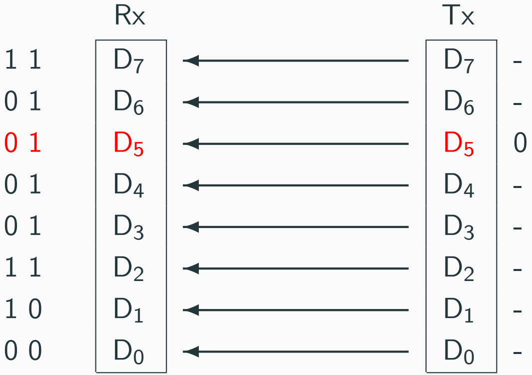
Example: 2 bytes of data are being transmitted in parallel

Parallel Transmission



Some interference on wire 5 means that the bit on that channel is delayed

Parallel Transmission



No data is interpreted as a 0,
and the late bit becomes mixed with the next byte

Parallel Transmission

By the time the final bit arrives,
we have 3 incorrect bytes of data

Parallel Transmission

- Limited to very short distances ($\approx 10\,\text{m}$) because of the problem of skewing
    - If we keep the cable short, less chance of problems

Serial Transmission

Single bits are transmitted one at a time,
only require a single wire

Serial Transmission

- Transmit bits one by one, therefore only need one wire
- Serial communications require parallel-to-serial and serial-to-parallel converters
    - Because sender and receiver usually handle data in 8-bit chunks
    - Often dealt with in software
- Two types:
    - Asynchronous - signalled start and stop of bytes
    - Synchronous - time-based

Asynchronous Transmission

- Timing of the bits isn't important
- Instead we use a *start* and *stop* signals
    - Specific bit patterns that **can't** be used in the data stream
    - Must be designed as part of your protocol
    - Prior agreement between the Tx and Rx
    - Start and stop signals must be different

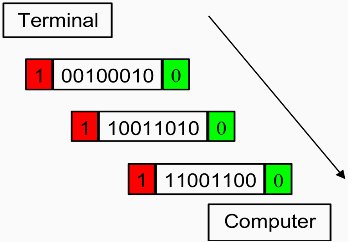
Asynchronous Transmission

Rx $\qquad\qquad\qquad\qquad$ Tx

S 1 0 1 1 1 0 1 0 S $\boxed{D_0} \longleftarrow \boxed{D_0}$ S 1 0 0 0 0 1 0 1 S

- Data is sent as follows:
  - Start signal alerts the receiver to the fact that data is coming
  - The byte of data
  - Stop signal (sometimes repeated, or with a gap afterwards) marks the end of the current byte

Asynchronous Transmission



- Start signals, stop signals, and gaps slow down the communication
- Cheap and effective if speed is not an issue
- Used to be used widely for communication between terminals and computers

Synchronous Transmission

- Timing of the bits is very important
- Data is transmitted in frames – units of (usually) many bytes
  - Bits within the frame are sent one after another without start/stop signals or gaps
  - It is the receiver's responsibility to separate the bit stream into usable bytes
  - The receiver has to know the speed of data transmission to work out where bytes start and end,
    i.e. the Tx and Rx have to be time-synchronised

Synchronous Transmission

	Rx		Tx	
1 0 1 1 1 0 1 0	$\boxed{D_0}$	$\longleftarrow$	$\boxed{D_0}$	1 0 0 0 0 1 0 1

Bits within the frame are sent one after another
without start/stop signals or gaps

Synchronous Transmission

- Data is transmitted in frames
- Timing of the bits is **very** important, because it's the receiver's responsibility to separate the bit stream into bytes
- Bits within the frame are sent one after another without start/stop signals or gaps

- Faster than asynchronous
- It is the basis of modern communications devices

The important things to remember:

- Know what bits and bytes are!
- ASCII is a standard coding system that uses 7-bits to denote a character
- Parity: odd, even, vertical, horizontal
  - Vertical parity uses a single extra bit to help spot errors
  - Horizontal parity allows us to pinpoint the error
- Transmission
  - Modes: simplex, half-duplex, full-duplex
  - Types: parallel, serial synchronous, serial asynchronous

Next time...

The Media part of Modes and Media
(probably after a lecture with Dave Price)